Yuhan Xu

Assignment 4

1. (a) $w = (X^T X)^{-1} X^T y = [0.9437\ 0.2137\ 0.2664\ -0.3922\ -0.0054\ -0.0176\ -0.1663\ -0.0823\ -0.1664]^T$

(b) we use training data X and w to calculate $X^T w$. If the result of $X^T w$ is positive, we classify the face image as happy; if the result is negative, then the image is angry.

(c) First feature, since it has the highest weight.

(d) I'll use features ①, ③, ④, which are 0.9437, 0.2664, -0.3922. The reason is that they have the highest weights among all nine features.
In this case, $\hat{w}$, which is the new classifier weights corresponding to of feature ①③④ is $\hat{w} = [0.7055\ 0.8738\ -0.7881]^T$.

(e) In 9 features case, there are 3 incorrect labels, so percent $\approx 2.34\%$
In 3 features case, there are 8 incorrect labels. So percent $= 6.25\%$
Matlab codes are shown at the end of assignment.

(f) Matlab codes are attached
average_rate = 0.4766

```
% Problem 1(e)
E = X*w; % for 9-features case
F = X_new*w_new; % for 3-features case

count = 0;
for i=1:length(y)
    if E(i)*y(i)<0
        count = count+1;
    end
end
disp(count/128);

count1 = 0;
for i=1:length(y)
    if F(i)*y(i)<0
        count1 = count1+1;
    end
end
disp(count1/128);

% Problem 1(f)
number_row = 0;
error_rates = 0;
X_new = X(:,[1 3:4]);

for i = 1:8
    testX = X_new(number_row+1:number_row+16,:);
    testy = y(number_row+1:number_row+16,:);
    trainingX = X_new([1:number_row number_row+17:128],:);
    trainingy = y([1:number_row number_row+17:128],:);
    w1 = ((transpose(trainingX)*trainingX)^-1)*transpose(trainingX)*trainingy;
    y_predict = testX*w1;
    number_row = number_row + 16;
    count = 0;
    for i = 1:16
        if y_predict(i)*y(i)<0
            count = count+1;
        end
    end
    misclassification = count/16;
    error_rates = error_rates + misclassification;
end

average_rate = error_rates/8;
disp(average_rate);
```

```
    0.0234

    0.0625

    0.4766
```

Published with MATLAB® R2016b